

11-04
158520

p. 16

Discrete Range Clustering Using Monte Carlo Methods

G. B. Chatterji and B. Sridhar

Quick Release – This Technical Memorandum is an unedited report. It is being released in this format to quickly provide the research community with important information.

March 1993

(NASA-TM-104004) DISCRETE RANGE
CLUSTERING USING MONTE CARLO
METHODS (NASA) 16 p

N93-24914

Unclass

G3/04 0158520



National Aeronautics and
Space Administration

Discrete Range Clustering Using Monte Carlo Methods

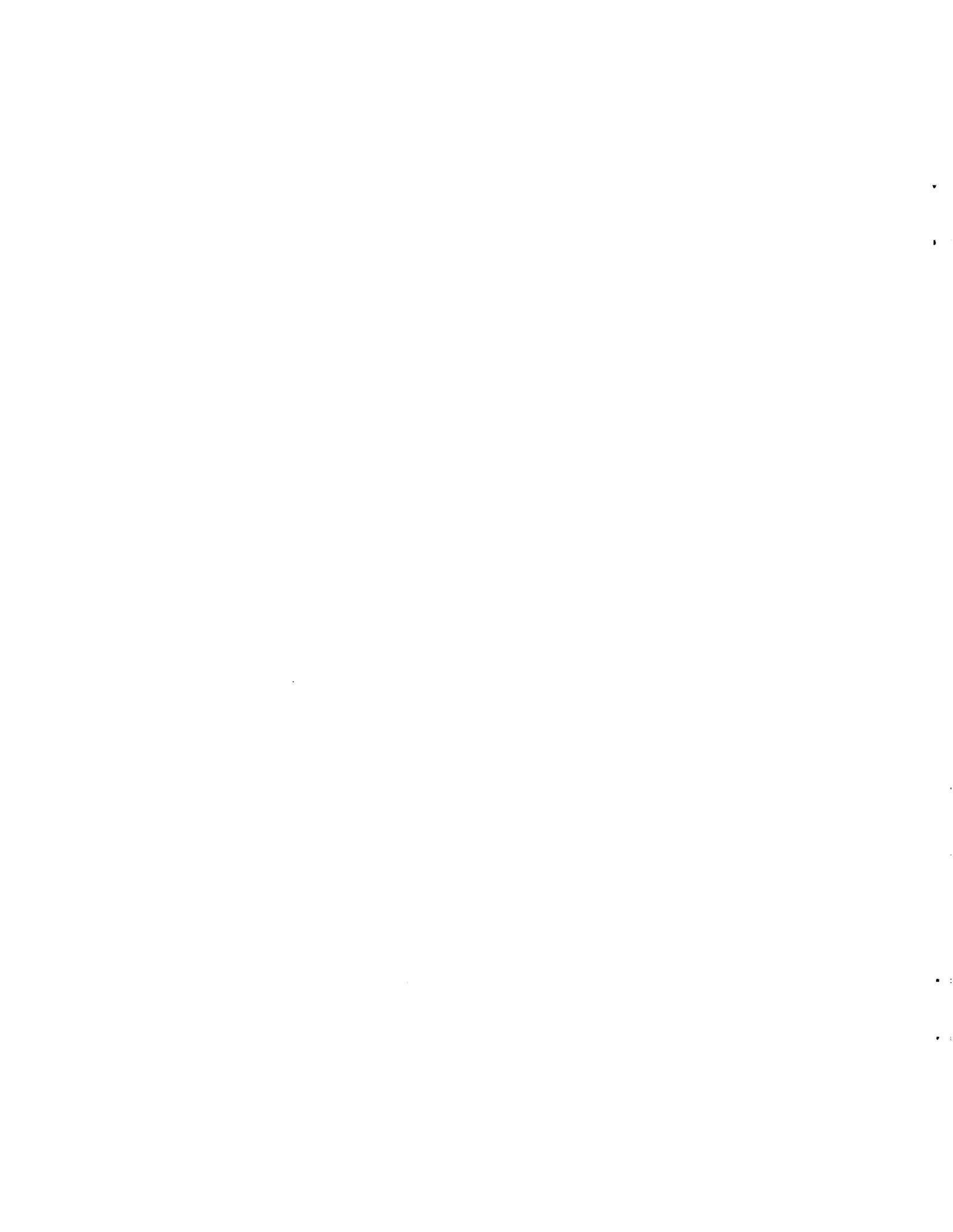
G. B. Chatterji and B. Sridhar, Ames Research Center, Moffett Field, California

March 1993



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000



SUMMARY

For automatic obstacle avoidance guidance during rotorcraft low altitude flight a reliable model of the nearby environment is needed. Such a model may be constructed by applying surface fitting techniques to the dense range map obtained by active sensing using radars. However, for covertness, passive sensing techniques using electro-optic sensors are desirable. As opposed to the dense range map obtained via active sensing, passive sensing algorithms produce reliable range at sparse locations and, therefore, surface fitting techniques to fill the gaps in the range measurement are not directly applicable. Both for automatic guidance and as a display for aiding the pilot, these discrete ranges need to be grouped into sets which correspond to objects in the nearby environment. The focus of this paper is on using Monte Carlo methods for clustering range points into meaningful groups. One of the aims of the paper is to explore whether Simulated Annealing methods offer significant advantage over the basic Monte Carlo method for this class of problems. We compare three different approaches and present results of application of these algorithms to a laboratory image sequence and a helicopter flight sequence.

1 INTRODUCTION

For vehicle guidance and navigation, it is always assumed that the position of all objects near the path of interest is available (see for example, refs. 1 and 2). The position of the objects is given by the constructed model of the neighboring environment. The first step in the model building process is sensing the environment using a sensor. In the case of an electro-optic sensor this begins with the acquisition of a sequence of images of the outside world. The second step involves processing the images to extract the position of the various objects in the field-of-view of the sensor. For example, optical flow/motion algorithms are used for extracting range to several locations using a sequence of images. The third step is to aggregate (cluster) the range at discrete locations into groups that represent objects in the environment. In this paper, we will focus on the third step.

Recently, several algorithms for computing depth, using a sequence of images, have appeared in the literature on computer vision (refs. 3-8). These algorithms

are successful in determining depth to only a few locations in the image plane due to various ambiguities. Sparse depth map represents discrete locations (features) in the sensors field-of-view where depth information is available. The range map created by these algorithms may be processed further to establish relationships between nearby features based on some measure of distance. Once the range features are aggregated into groups, the range map may be partitioned to represent the objects in the nearby environment. In the subsequent sections we will focus on exploring statistical methods for achieving this partition.

One of the ways of solving the clustering problem is to cast it as a discrete optimization problem, which minimizes a cost function to partition the depth map into groups of features. Monte Carlo method and its various modified forms, also known as Simulated Annealing methods, may then be used to optimize the cost. In reference 9, Simulated Annealing has been used for assigning features to a pre-defined number of image regions by minimizing the sum of within group variances. To start the process, an image is segmented and labeled into regions. Next, features are assigned to the regions in the image plane. Features are then reassigned to the regions and group properties such as, mean depth and variance, are computed. Simulated Annealing is then used to iteratively minimize the sum of group variances, thus achieving the final configuration. It may be noted that a structure was first imposed by fixing the number of groups. Given this structure, the algorithm was used to determine the optimal group membership (which feature belongs to which group) that fit this structure.

In unsupervised clustering, neither the number of groups, nor the memberships are known and therefore, as opposed to the approach taken in reference 9, we do not fix the number of groups but iteratively modify the structure by adding new groups and optimize the cost at each such step. The question of an appropriate stopping criteria is discussed later. In the next section we discuss the Monte Carlo methods. In section 3 the laboratory and the flight image sequences are described. Subsequently, in section 4, the process of initial grouping is described. Section 5 describes the Simulated Annealing, modified Simulated Annealing and the Monte Carlo method for minimizing a cost function to achieve the optimal grouping. The results of application of these algorithms to a laboratory and a flight sequence are described in section 6. Finally, conclusions are drawn in section 7.

2 MONTE CARLO METHODS

Interest in the modified forms of the Monte Carlo method seems to have been stimulated by reference 10, in which a modified form of the Monte Carlo method was proposed for investigating equilibrium properties of liquids. By conventional numerical methods this would involve solving several hundred-dimensional integrals. Alternatively, the equilibrium property may be computed by using the Monte Carlo method for multi-dimensional integrals by integrating over a random sampling of points instead of over a regular array of points. The modified Monte Carlo approach suggested in reference 10 consists of choosing the configurations with a probability $\exp(-E/kT)$ and weighting them evenly. Here, E is the energy of the configuration, k is the Boltzmann constant and T is the temperature. Their algorithm for accomplishing this may be summarized as follows: Initially N particles are placed in any configuration, for example, in a regular lattice. Then a particle is moved such that it is equally likely to be anywhere within a square centered about its original position. Change in the energy of the system, ΔE , is calculated for this move. If $\Delta E < 0$, i.e., if the move brings the system to a state of lower energy, the move is allowed and the particle is put in its new position. If $\Delta E > 0$, the move is allowed with a probability $\exp(-\Delta E/kT)$, i.e., if a random number ζ between 0 and 1 is less than $\exp(-\Delta E/kT)$, the move is allowed. If $\zeta > \exp(-\Delta E/kT)$, the particle is returned to its old position. Average property is then computed based on the resulting configuration, whether it changed or not, due to the move. The procedure then proceeds to consider the next particle and so on.

A later modification in reference 11 uses the algorithm of reference 10 with a decreasing temperature schedule. This algorithm is known as the Simulated Annealing algorithm. The temperature scheduling idea is motivated by the analogy to the annealing process in metals. The Simulated Annealing algorithm may be summarized as follows: Start with an initial configuration at a certain temperature and make a random move. Due to this move, compute the change in energy, ΔE . If $\Delta E < 0$, accept the change. If $\Delta E > 0$, accept the change with a probability $\exp(-\Delta E/T)$. In other words, if $\Delta E > 0$, generate a uniform random number ζ between 0 and 1 and check if $\zeta < \exp(-\Delta E/T)$. If the inequality is satisfied, accept the new configuration else, preserve the old configuration. Repeat the above steps a number of times, n , and then lower

the temperature by a factor $0 < k < 1$. Start from this configuration and re-do the steps till the change in energy is small. The details of the algorithm are available in references 12 and 13. The main implication of temperature scheduling is that at a higher temperature, increase in ΔE is more freely allowed than at a lower temperature.

Usually, it is claimed that the Simulated Annealing technique has the ability to "climb out" (because it allows $\Delta E > 0$) of the local minimum (refs. 12-14). This, however, is misleading. The main feature which allows the algorithm to find the global minimum is by randomly choosing starting locations within the search space. These ideas may be clarified by the following one-dimensional search example shown in figure 1.

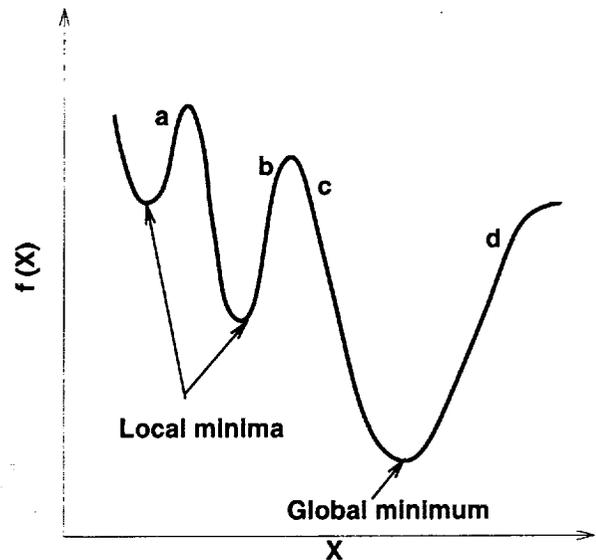


Figure 1. Search in one-dimension.

The objective of the one-dimensional search problem (see fig. 1) is to determine the value of x for which $f(x)$ is a global minimum. It is assumed that $f(x)$ is continuous. Let us say that we have an algorithm which climbs over the hill. Such an algorithm would eventually find the minimum, but it would probably take the same amount of time as an algorithm that systematically explores the entire domain of x . Let us now say that we have an algorithm that starts out with random initial conditions (for example: locations a, b, c, and d in fig. 1) and uses a gradient search method to locate a minimum. This algorithm will succeed in reaching the global minimum in very few steps.

The algorithm in reference 10 was introduced to solve a specific type of problem where it made good sense. Since its introduction, the Simulated Annealing method has been proposed for solving many different kinds of minimization problems. One of the goals of this paper is to determine whether the Simulated Annealing methods offer significant advantages over the basic Monte Carlo method for partitioning the sparse depth map into objects. For the comparison we have applied the algorithms to range maps generated from both, a laboratory image sequence and a flight sequence.

In the next section we describe the data sets.

3 DATA SETS

In this section we will first describe the laboratory image sequence and then the flight sequence.

The laboratory image sequence used in this work consists of 80 images that were acquired by a camera mounted on a 3 degree-of-freedom motion table. Figure 2 shows the first image and figure 3 the last image in the sequence. The objects in the view of the imaging sensor are labeled from A-L in figure 2 where, A is the tape on the back wall, B is the left pencil, C is the soda can, D is the wire, E is the right pencil, F is the soda can base, G is the table, H is the tape on the table in front of the soda can base, I is the bracket, J is the tape on the bracket, K is the tape on the table behind and to the left of the bracket, and L is the tape on the table to the left of the bracket. The details of the data acquisition process for the laboratory image sequence are described in reference 15.

The flight image sequence consists of 240 images which were acquired by a camera mounted under the rotorcraft nose and oriented roughly in the direction of the flight so that the designated obstacles could be observed. The position of the camera and its orientation with respect to the rotorcraft were held constant throughout the flight. The 45th image is shown in figure 4 and the 60th image in figure 5. The objects in the view of the imaging sensor are labeled from A-H in figure 4 where, A, B, C, D, and E are trucks on the runway, F is the time stamp, G is the runway, and H is the rotorcraft nose boom. The trucks A, B, and C are arranged with A being closest to the camera and C being farthest with B in between. Truck D is between trucks A and B, and truck E is between trucks B and C. The details of the methodology used to develop

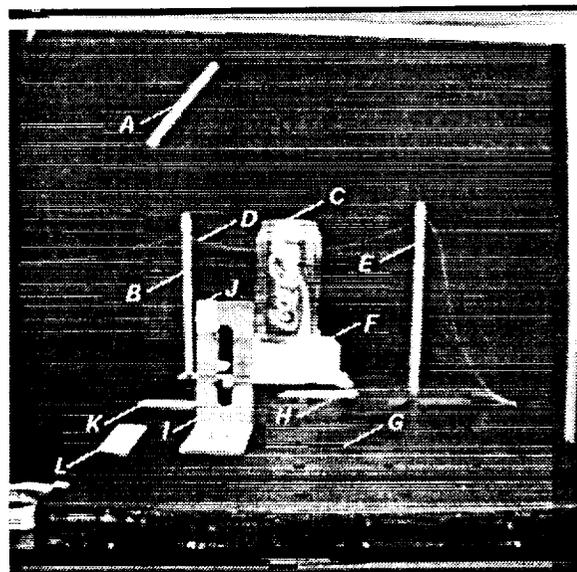


Figure 2. First laboratory image.

the flight data base consisting of imagery, rotorcraft and sensor parameters, and ground-truth range measurements is described in reference 16.

The method of depth computation consisted of feature tracking based on correlation followed by recursive depth estimation using an Extended Kalman Filter (refs. 3, and 6-8). The ranging algorithm outputs a sparse depth map (displayed as white squares in

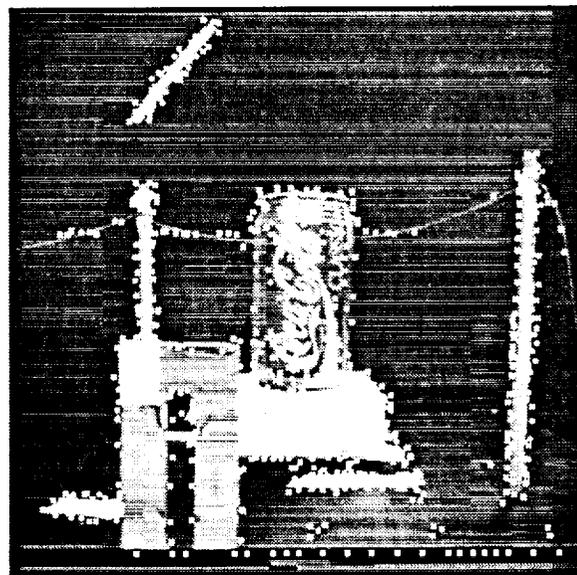


Figure 3. Last laboratory image.

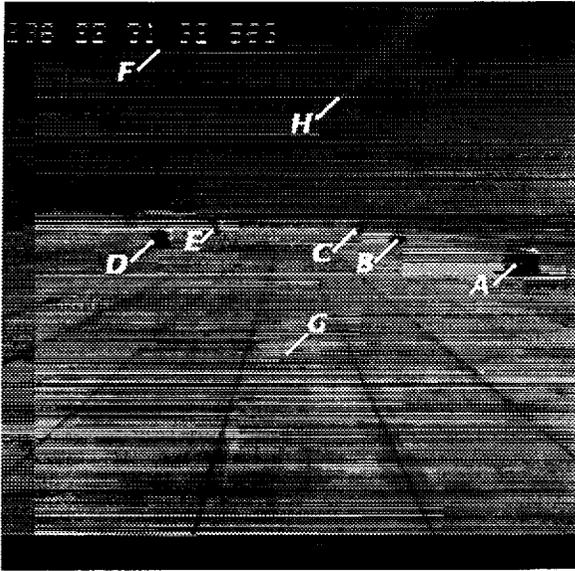


Figure 4. 45th flight image.

figs. 3 and 5). The depth map consists of a list of features characterized by a “u,” “v” location in the image plane, an identification tag and computed depth. The u, v locations and depth are directly related to the x, y, and z locations in the inertial frame via the inverse perspective projection equations, the camera-to-body and

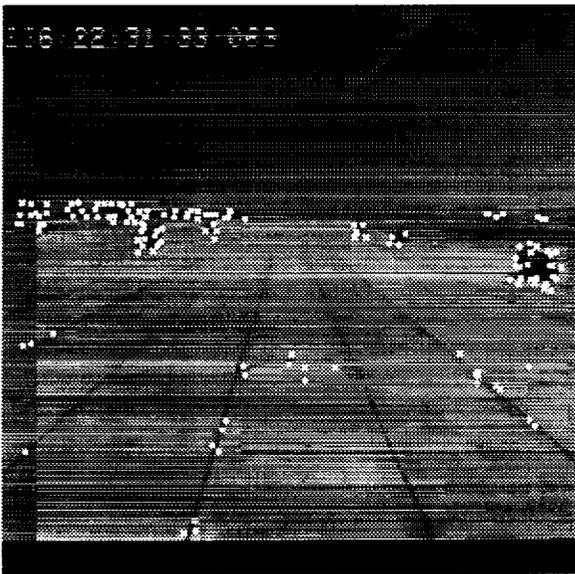


Figure 5. 60th flight image.

body-to-inertial transformation matrices. The identification tag identifies the feature through a series of images.

The depth map in figure 5 is for those features which existed in at least 10 of the 15 depth maps processed, i.e., the presence of the same identification tag was checked in 15 depth maps and if it existed in 10 of them, it was selected. These depth maps correspond to the 45th image through the 60th image. One of the advantages of doing this pre-processing is that new or unconverged features are eliminated. This results in a higher confidence depth map.

4 INITIAL GROUPING

Let us assume that the depth values of the features, corresponding to the viewed objects, have a gaussian distribution. The objective now is to describe the depth data by a combination of gaussians. The first step in this process is the construction of a depth histogram with the number of features as a function of depth. This process may be summarized as follows:

- Compute the minimum depth, d_{min} .
- Compute the number of bins, n_{bins} as,

$$n_{bins} = \frac{d_{max} - d_{min}}{bin_size} \quad (1)$$

where, d_{max} is the maximum depth of interest and bin size is the depth resolution. The maximum observable depth and depth resolution may be specified in terms of the camera parameters and the stereo or motion baseline.

- Count the number of features within each bin. This corresponds to the frequency at a depth corresponding to the center of the bin.
- Compute the maximum frequency and normalize all frequencies with respect to it.

This results in a normalized histogram sampled at bin size intervals. For the depth map shown in figure 3, the depth histogram is shown in figure 6. Here, the bin size was chosen to be 2 inches. For the flight sequence depth map shown in figure 5, the depth histogram is shown in figure 7. The bin size in this case was 20 feet.

The second step in the grouping process is detection of the peaks of the histogram. A peak is defined

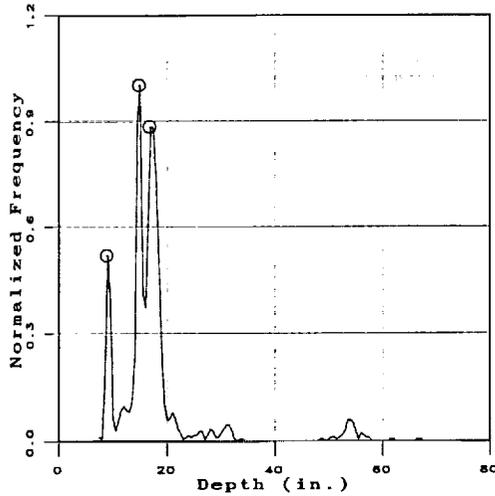


Figure 6. Depth histogram for laboratory sequence.

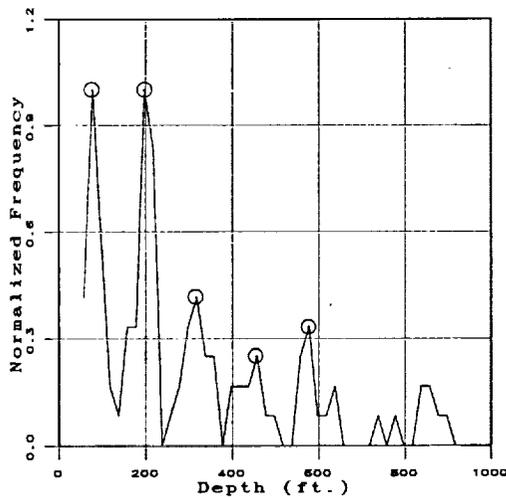


Figure 7. Depth histogram for flight sequence.

as a maximum bounded on both sides (left and right) by minima such that the difference between the peak and the valleys exceeds a threshold, "peakiness." The exact implementation of the peak detection algorithm is fairly elaborate and will not be discussed here. The central idea however, is to determine a peak by bracketing it between a proper left minimum and a proper right minimum such that the peakiness criteria is satisfied. A proper minimum is determined by bracketing the minimum between two peaks such that the peakiness criteria is satisfied with respect to the bounding

peaks. For the histograms in figures 6 and 7, the circles around the peaks in figures 6 and 7 show the detected peaks. The peakiness value used was 0.1.

Next, we approximate the histogram as a sum of m gaussians. The approximation to the histogram is achieved by minimizing the sum of the squares of the error defined by equation (3).

$$\text{minimize } \sum_{i=1}^n e_i^2 \quad (2)$$

where

$$e_i = \zeta_i - \sum_{j=1}^m k_j e^{-\frac{(\delta_i - \mu_j)^2}{2\sigma_j^2}} \quad (3)$$

Here, k_j is the scale factor, μ_j is the mean and σ_j is the standard deviation of j^{th} gaussian. At n depth locations, the depth is δ_i and the normalized frequency is ζ_i . For minimization, a MINPACK (ref. 17) routine LMDIF1 is used. The routine LMDIF1 is a modified version of the Levenberg-Marquardt algorithm. Based on the detected peaks, an initial estimate of the k_j , μ_j and σ_j for each gaussian is provided to the minimization routine. The individual gaussians approximating the depth histogram (in fig. 6) are shown in figure 8. Similarly, the five gaussians approximating the flight sequence depth histogram (see fig. 7) are shown in figure 9.

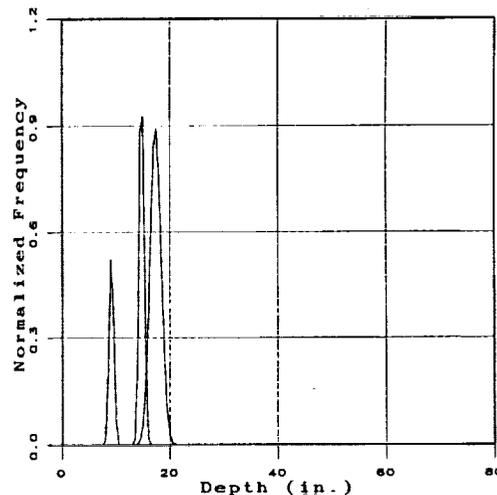


Figure 8. Gaussians approximating the histogram for the laboratory sequence.

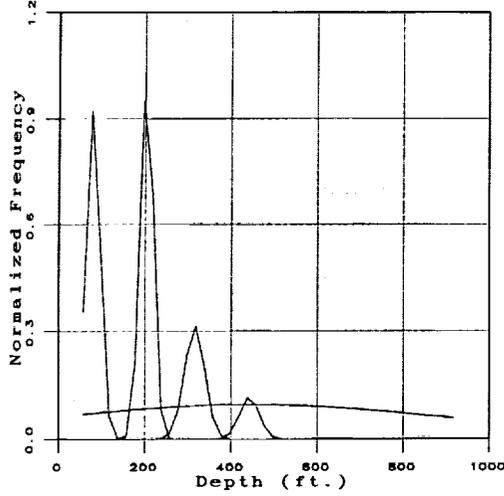


Figure 9. Gaussians approximating the histogram for the flight sequence.

Finally, the features are assigned to the groups represented by the gaussians. For this purpose, consider two gaussians next to each other. In order for them to intersect, the following relationship must hold:

$$k_1 e^{-\frac{(\delta-\mu_1)^2}{2\sigma_1^2}} = k_2 e^{-\frac{(\delta-\mu_2)^2}{2\sigma_2^2}} \quad (4)$$

This simplifies to:

$$(\sigma_2^2 - \sigma_1^2)\delta^2 - 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)\delta + (\mu_1^2\sigma_2^2 - \mu_2^2\sigma_1^2) - 2\sigma_1^2\sigma_2^2 \ln\left(\frac{k_1}{k_2}\right) = 0 \quad (5)$$

This equation may be solved to obtain the depth δ at which the two gaussians intersect. It may be noted that if the standard deviations σ_1 and σ_2 are equal, equation (5) is reduced to a linear equation. The intersection depth marks the outer limit of gaussian 1, and inner limit of gaussian 2. By repeating this procedure, inner and outer limits are determined for each gaussian. The minimum depth marks the inner limit for the 1st gaussian, and the maximum depth marks the outer limit of the last gaussian. Each feature is allocated to one of the gaussians based on whether the depth corresponding to the feature lies within the inner and outer limits of that gaussian. The resulting grouping is coded in a matrix, \mathbf{G} where the number of columns is equal to the number of groups, and the identification tags of all features belonging to a particular group are stored in a single column. The number of members

in each group is coded in a vector, \mathbf{N} . This way each member of the group is accessible via the vector \mathbf{N} and the matrix \mathbf{G} .

In the next section we describe grouping based on the basic Monte Carlo method and its modified versions.

5 OPTIMAL GROUPING

Starting with the initial grouping described in the previous section, the goal now is to re-allocate the features to various groups such that a cost function is minimized. Several criterion functions for clustering are described in reference 18. Out of these, we have chosen the cost function to be the ratio of the trace of the within group scatter matrix to the trace of the between groups scatter matrix:

$$J = \frac{\text{tr}(S_W)}{\text{tr}(S_B)} \quad (6)$$

$$S_W = \sum_{i=1}^c \sum_{j=1}^{n_i} (x_j^i - m_i)(x_j^i - m_i)^t \quad (7)$$

$$S_B = \sum_{i=1}^c n_i(m_i - m)(m_i - m)^t \quad (8)$$

Here, c is the number of groups, n_i is the number of features in the i^{th} group, m_i is the mean of the features in the i^{th} group, m is the mean of all features, x_j^i is the j^{th} feature in the i^{th} group, S_W is the within group scatter matrix, S_B is the between groups scatter matrix and J is the objective function. The objective function, J , can be minimized using discrete optimization techniques. Minimizing J has the implication of decreasing the within cluster distance and increasing the between cluster distance.

We first describe the Simulated Annealing algorithm. Next, we describe the basic Monte Carlo method. Finally, we describe the modified Simulated Annealing algorithm.

The Simulated Annealing algorithm for refining the grouping may be summarized as follows:

1. Compute the cost, J , for the initial grouping using equation (6) and initialize the temperature, T , to a large number.
2. While ($T > \beta$) execute the following steps or exit the algorithm.

3. Set the iteration counter, $I = 1$.
4. Randomly select a feature, F , such that every feature has an equal probability of being selected.
5. Randomly select a group, g , such that every group has an equal probability of being selected.
6. Check the grouping matrix, \mathbf{G} , to see if F is already a member of g . If yes, go to step 10. If no, proceed to the next step.
7. Remove F from its group and add it to the g group. This results in a modified grouping matrix, $\hat{\mathbf{G}}$, and the membership vector $\hat{\mathbf{N}}$.
8. Evaluate the cost, \hat{J} , for grouping $\hat{\mathbf{G}}$ using equation (6).
9. Compute $\Delta J = \hat{J} - J$. If $\Delta J < 0$, set: $\mathbf{N} = \hat{\mathbf{N}}$, $\mathbf{G} = \hat{\mathbf{G}}$ and $J = \hat{J}$. If $\Delta J > 0$ and a uniform random number between 0 and 1, $\tau < \exp(-\Delta J/T)$, then set: $\mathbf{N} = \hat{\mathbf{N}}$, $\mathbf{G} = \hat{\mathbf{G}}$ and $J = \hat{J}$.
10. If $I > I_{max}$ or if $\frac{|\Delta J|}{J} < \varepsilon$ go to step 11, else increment the iteration counter, $I = I + 1$ and go back to step 4.
11. Reduce the temperature, T , by a factor $0 < \alpha < 1$; $T = \alpha T$.
12. Go back to step 2.

Here, I_{max} is the maximum number of iterations, ε is a small number for checking relative convergence, α is the cooling schedule parameter and β is a small number used for exiting the algorithm. Convergence criteria in step 10 may also be based on absolute convergence, $|\Delta J| < \varepsilon$. Generally, absolute criteria will require a large number of iterations. An additional criteria requiring that a certain small number of trails be made before going to step 11 is a useful one due to the fact that the same feature F and group g may be selected successively. It may be noted that the convergence and rate of convergence depend on the initial temperature, T , and the cooling schedule parameter, α . For the clustering problem of the type discussed in this paper, a high initial temperature T may cause the solution to diverge. This is due to the reason that at very high temperature, many configurations which increase the optimization cost will be permitted thereby, altering the group properties (for example, means) to an extent that it may no longer be possible to recover

the structure. To ensure that most of the time $\Delta J < 0$, an appropriate initial temperature must be chosen. The cooling schedule parameter α is directly related to the number of iterations.

The basic Monte Carlo algorithm may be summarized as follows:

1. Compute the cost, J , for the initial grouping using equation (6).
2. Set the iteration counter, $I = 1$.
3. Randomly select a feature, F , such that every feature has an equal probability of being selected.
4. Randomly select a group, g , such that every group has an equal probability of being selected.
5. Check the grouping matrix, \mathbf{G} , to see if F is already a member of g . If yes, go to step 9. If no, proceed to the next step.
6. Remove F from its group and add it to the g group. This results in a modified grouping matrix, $\hat{\mathbf{G}}$, and the membership vector $\hat{\mathbf{N}}$.
7. Evaluate the cost, \hat{J} , for grouping $\hat{\mathbf{G}}$ using equation (6).
8. Compute $\Delta J = \hat{J} - J$. If $\Delta J < 0$, set: $\mathbf{N} = \hat{\mathbf{N}}$, $\mathbf{G} = \hat{\mathbf{G}}$ and $J = \hat{J}$.
9. If $I > I_{max}$ or if $\frac{|\Delta J|}{J} < \varepsilon$ exit or increment the iteration counter, $I = I + 1$ and go back to step 4.

It may be seen that most of the steps of the Simulated Annealing and the basic Monte Carlo algorithms are alike. The differences are the following: In the basic Monte Carlo method, temperature scheduling is not used and the cost is not allowed to increase (compare step 8 of the Monte Carlo algorithm with step 9 of the Simulated Annealing algorithm).

A modified version of the Simulated Annealing approach is suggested in reference 19. The difference is in step 9 of the Simulated Annealing algorithm, which is:

9. Compute $\Delta J = \hat{J} - J$. If $\Delta J < 0$, set: $\mathbf{N} = \hat{\mathbf{N}}$, $\mathbf{G} = \hat{\mathbf{G}}$ and $J = \hat{J}$. If $\Delta J > 0$ and a uniform random number between 0 and 1, $\tau < \exp(-\Delta J/(J \times T))$, then set: $\mathbf{N} = \hat{\mathbf{N}}$, $\mathbf{G} = \hat{\mathbf{G}}$ and $J = \hat{J}$.

Note, inclusion of the cost, J in the function $\exp(-\Delta J/(J \times T))$. This has a similar effect as the temperature, T . For large J , configurations which increase the cost are permitted, and as the cost decreases, fewer cost increasing configurations are allowed. Some of these features are illustrated via examples given below.

Figure 10 shows the optimization cost, for the laboratory image sequence, as a function of the number of iterations. The graphs with legends MC, SA, and MSA show the cost reduction achieved by the basic Monte Carlo, Simulated Annealing and the modified Simulated Annealing methods, respectively. In each case, the optimization was started with 3 groups shown in figure 8. For the flight sequence, the de-

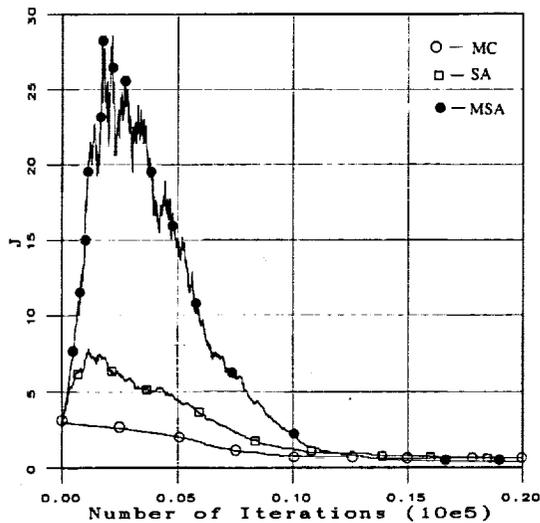


Figure 10. Cost for the laboratory sequence.

crease in cost achieved by the basic Monte Carlo (MC), the Simulated Annealing (SA), and the modified Simulated Annealing (MSA) methods is shown in figure 11. The initial grouping for the flight sequence consisted of 5 groups shown in figure 9. Both, in the case of optimization using the laboratory and flight sequence, it was also ensured that the total number of random moves for the three algorithms was the same. The cost function (eq. 6) was computed via equations (7) and (8), by defining x as feature coordinates in a two-dimensional inertial frame on horizontal plane. This means that the altitude component of the position vector was not used for grouping. The initial temperature

for the Simulated Annealing methods was chosen to be $T = 0.1$. It may be seen from figures 10 and 11 that the basic Monte Carlo method reduces the cost faster than the Simulated Annealing methods. Initially, both the Simulated Annealing methods allow cost increasing configurations and eventually converge to nearly the same cost.

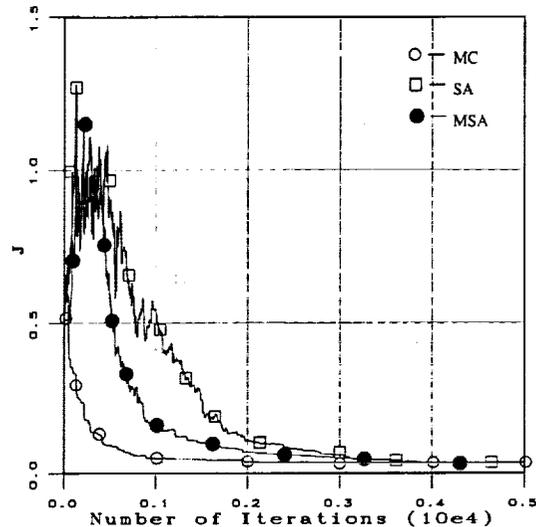


Figure 11. Cost for the flight sequence.

It was observed in reference 19 that the standard Simulated Annealing method requires a large number of steps to reach the global minimum when compared to the modified Simulated Annealing method. One may compare figures 10 and 11 to see that the number of steps required for convergence in the case of modified Simulated Annealing depends on the numerical value of the cost. It is further suggested in reference 19 that the cost keeps increasing and decreasing thereby not providing a logical termination criterion. It appears that the final temperature in the case of Simulated Annealing and the product of the final cost and temperature in the case of modified Simulated Annealing determine whether the cost keeps increasing and decreasing or settles down. It may be seen from figures 10 and 11 that for a final temperature of $T < 0.0001$ the cost settles down for both the methods. In conclusion, both the Simulated Annealing methods may be made to behave in a similar manner by choosing the appropriate initial temperatures.

So far we have described three optimization methods for a fixed number of groups. These algorithms result in a grouping G , which is optimal with respect to the cost J for the number of groups, c . The number of groups, c is determined by the initial grouping algorithm described in the previous section. This means that for a fixed structure (number of groups), an optimal feature grouping (which feature belongs to which group) may be achieved. We now describe a method for creating new groups and applying the optimization methods to achieve optimum number of groups and optimal group membership.

One of the ways of creating a new group, which is away (not close to any particular group) from all the groups, is to do the following: For each feature, compute the Euclidean distance to the mean of each group. Select the minimum distance. Once this is done for all the features, select the feature whose minimum distance to the groups is a maximum. The selected feature is the pivot for the new group. Features, which are closest to this pivot feature, are allocated to this group thus a new grouping matrix G and a membership vector N are created to reflect the new grouping.

The optimization methods described before may now be re-applied to the new grouping to result in an optimal grouping. It is hoped that by successively applying the optimization algorithm and the new group creation algorithm a stage will be reached where addition of a new group will not result in a significant decrease in the optimization cost. We will call the grouping at this stage to be optimal.

Starting with the initial grouping (with cost $J = 2.997$) for the laboratory sequence, shown in figure 8, the optimization algorithms were applied to minimize the cost. Next, a new group was created by using the new group creation algorithm and the grouping was altered. The optimization methods were re-applied to this configuration. Results of iterative application of the optimization algorithms and new group creation algorithm are shown in figure 12. The graphs, MC, SA, and MSA depict the cost reduction achieved by the basic Monte Carlo, Simulated Annealing, and the modified Simulated Annealing methods. The results of application of the iterative group creation and optimization method for the flight sequence is shown in figure 13. As before, graphs with legends MC, SA, and MSA show the cost reduction achieved by the basic Monte Carlo, Simulated Annealing, and the modified Simulated Annealing methods. Initially, 5 groups shown in figure 9 were input for optimization. A cost

of 0.5143 was computed for the initial grouping. The symbol with legend H & S, in figure 13, shows the minimum cost for 10 groups using the algorithm in reference 9.

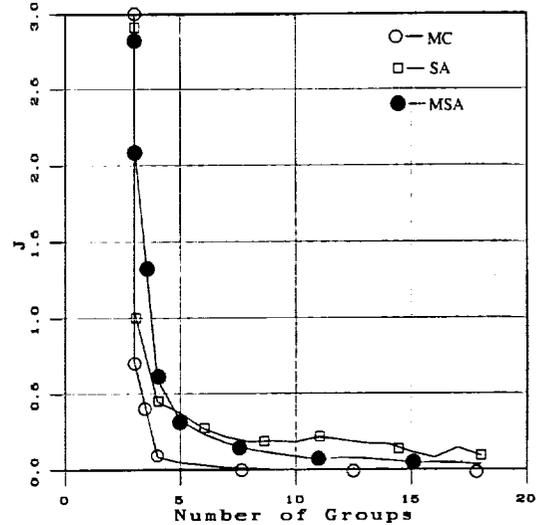


Figure 12. Cost for the laboratory sequence.

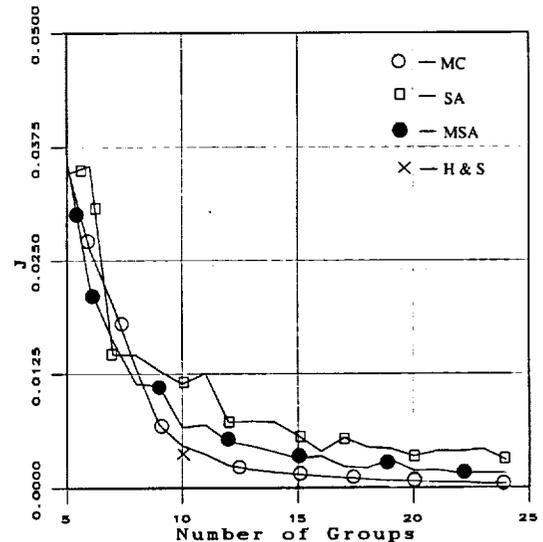


Figure 13. Cost for the flight sequence.

For optimization at each stage (for a fixed number of groups) 12,000 iterations were done for the laboratory sequence, and 5,000 iterations were done for the flight sequence. The initial temperature for the Simulated Annealing methods was chosen to be $T = 0.1$.

It may be seen from figures 12 and 13 that the basic Monte Carlo method reduces the cost faster than the Simulated Annealing methods and arrives at the optimal solution, which partitions the initial 3 groups into 8 groups for the laboratory sequence and 5 groups into 24 groups for the flight sequence. In the next section we describe the optimal grouping results based on these methods.

6 RESULTS

For the laboratory sequence depth map shown in figure 6, a histogram was constructed and the peaks of the histogram were detected by the peak detection algorithm (see fig. 6). Next, the histogram was approximated as a sum of gaussians. Starting with an initial estimate of the mean, standard deviation and the scale factor for each gaussian, the sum of squares of the fit error (between the histogram and the sum of gaussians) was minimized to obtain improved estimates of the mean, standard deviation and the scale factor for the gaussians. The three gaussians approximating the histogram are shown in figure 8. Finally, each feature was assigned to one of the gaussians based on whether its depth was between the inner and outer limits of the gaussian. The resulting grouping is shown in figure 14. Boundaries have been drawn around features to show that they belong to the same group. It may be seen from the figure that most of the features on the bracket (see fig. 2, label I), tape on the bracket (J), and the edge of the table are classified as group 1 (corresponding to the 1st gaussian). The right pencil (E), the tape on the table in front of the soda can (H), a part of the wire (D), few features on the base of the soda can (F), few features on the table (G), and a few features on the tape on the table to left of the bracket (K) are classified as group 2 (corresponding to the 2nd gaussian). Most of the soda can (C), the soda can base (F), the wire (D), the left pencil (B), the tape on the back wall (A), the tape on the table to the left of the bracket (K), and a few features on the tape on the table in front of the soda can base (H) are classified as group 3 (corresponding to the 3rd gaussian). It may

be seen that small groups of features appear within or close to larger groups.

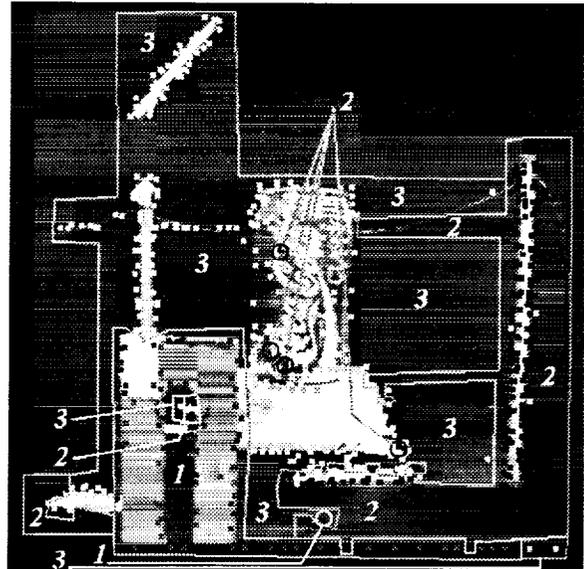


Figure 14. Initial grouping based on fitting gaussians.

Starting with the initial grouping (one-dimensional clustering) described in figure 14, the basic Monte Carlo method was applied to re-group the features based on the distance measure in the horizontal inertial plane (two-dimensional clustering). To minimize the cost, 12,000 random moves (for example, see fig. 10) were made. The grouping matrix, G , and the vector, N , were updated to reflect this grouping. A new group was then created by following the new group creation algorithm. The process of optimization and new group creation was repeated till each group had at least three features. The final grouping consisting of 11 groups is described in figure 15.

In this figure boundaries have been drawn around features to show that they belong to the same group. The outliers are demarcated by circles. It may be seen from figure 15 that group 1 consists of the bracket (see fig. 2, label I), tape on the bracket (J), and the edge of the table. Group 2 consists of the right pencil (E), a portion of the wire (D), a part of the table (G), and a portion of the tape on the table (H), which is in front of the soda can. The outliers of group 2 lie on the base of the soda can (F). Features corresponding to the tape on the back wall (A) belong to groups 3 and 4. Group 5 consists primarily of features belonging to the

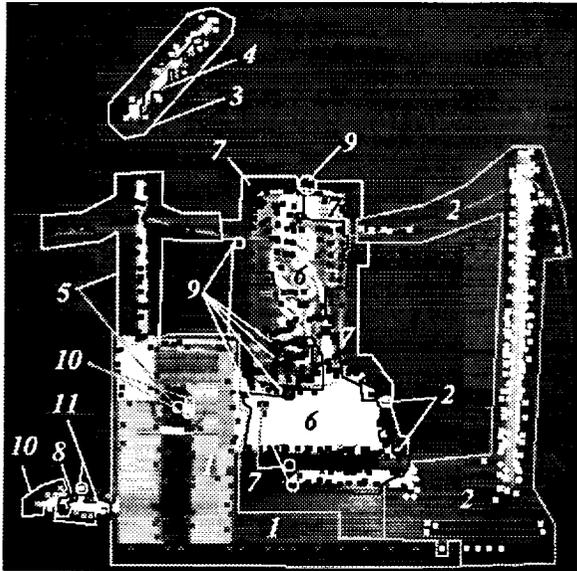


Figure 15. Optimal grouping for laboratory sequence.

left pencil (B). One outlier of group 5 lies close to the bracket (I). Groups 6 and 7 are made up of features that belong to the soda can (C) and the base of the soda can (F). It may be seen that features on the soda can (C) and the soda can base (F) are mostly grouped in the 6th group. A smaller portion of the features on the soda can are members of group 7. Groups 8, 10, and 11 are made up of features that belong to the tape on the table (K). Most of the members of group 9 lie on the soda can (C). This completes our discussion of the laboratory image sequence. Next, we describe the results of application of the initial and optimal grouping algorithms to the flight image sequence.

For initial grouping of the flight image sequence range features, depth histogram, shown in figure 7 was approximated by 5 gaussians, shown in figure 9. Features were then allocated to the groups represented by the 5 gaussians. This is shown in figure 16. Most of the features on the runway (see fig. 4, label G) are members of group 1. The features on truck A belong to group 2. Four outliers of group 2 are on the runway (G). Group 3 is mostly composed of features on the trucks D and E and one feature each on the trucks B and C. Few features on truck D and on the runway (G) form the 4th group. The 5th group consists of features on the trucks B, C, and E, and the background. Two outliers of group 5 lie on the truck D. The symbol FOE shows the location of the focus of expansion on this image.

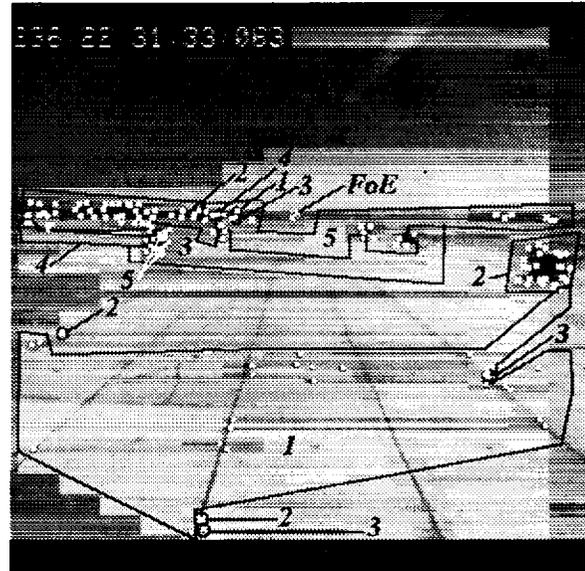


Figure 16. Initial grouping for flight sequence.

Starting with the initial grouping, described above, optimization using the basic Monte Carlo method and new group creation were done iteratively to arrive at the optimal grouping. Groups with less than three features were removed. This resulted in 12 groups. The last 2 groups consist of features from the background so the 10 closest groups are shown in figure 17. Groups 1-5 and 7 are composed of features on the runway (G). The 6th group is mostly composed of features on truck A. Groups 8 and 9 contain features of the background and features on trucks D and E, respectively. Finally, features on trucks B and C belong to group 10.

The salient features of the optimal grouping are the following: The features which are close are grouped together (for example, see group 2 in fig. 15). The left pencil (B) is no longer grouped with the soda can (C). The portion of the tape on the table (H), which is close to the soda can, is grouped together with the soda can (C). Similarly for the flight sequence, features that are closer to trucks A, D, and E are grouped with trucks A, D, and E, respectively. It may be seen in figure 17 that the outliers of initial groups (for example, see group 2 in fig. 16) are merged with the features (see groups 1, 4, 5, and 9 in fig. 17) that they were close to form optimal groups.

It may be seen from figures 15 and 17 that some groups span across a large number of pixels in the image plane therefore, suitable cluster density measures

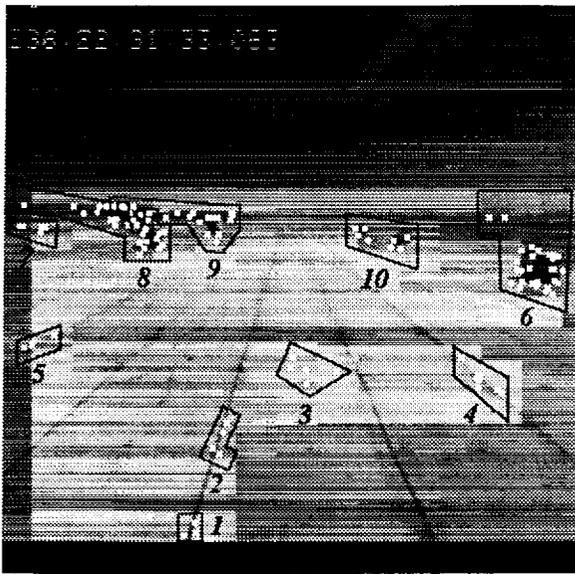


Figure 17. Optimal grouping for flight sequence.

(for example, see ref. 20) may be used in the image plane to further refine the grouping. It is quite possible that the final result of all the techniques namely, initial grouping, optimal grouping in the horizontal plane and re-grouping in the image plane, may result in an over-segmented depth map. From rotorcraft low altitude guidance point of view a reasonably oversegmented depth map may not pose a significant problem. An undersegmented depth map, on the other hand, may not be acceptable because in this case one would end up interpolating across various physical objects, thus blocking potential openings.

7 CONCLUSIONS

Starting with a depth map, computed by an optical flow method for a laboratory image sequence and a flight image sequence, a method based on fitting the sum of gaussians to the depth histogram was described to form the initial groups. Several feature grouping techniques based on the basic Monte Carlo method and its modified forms (Simulated Annealing methods) were then examined to refine this grouping based on separation in the horizontal inertial plane. For the example depth map, convergence characteristics, and the optimization cost as a function of the number of groups were examined for the various methods. Additionally,

it was shown that the basic Monte Carlo method converged faster and lowered the cost compared to the Simulated Annealing methods. Finally, the limitations of the resulting grouping are pointed out.

REFERENCES

1. Cheng, V. H. L.: Concept Development of Automatic Guidance for Rotorcraft Obstacle Avoidance. IEEE Transactions on Robotics and Automation, vol. 6, no. 2, April 1990, pp. 252-257.
2. Cheng, V. H. L.; and Sridhar, B.: Considerations for Automated Nap-of-the-Earth Rotorcraft Flight. J. American Helicopter Society, vol. 36, no. 2, April 1991, pp. 61-69.
3. Sridhar, B.; Suorsa, R.; and Hussien, B.: Passive Range Estimation for Rotorcraft Low Altitude Flight. To appear in Machine Vision and Applications, 1991.
4. Menon, P. K. A.; Chatterji, G. B.; and Sridhar, B.: Passive Obstacle Location for Rotorcraft Guidance. AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, August 12-14, 1991.
5. Skifstadt, K.; and Jain, R.: Range Estimation from Intensity Gradient Analysis. Machine Vision and Applications, vol. 2, no. 1, 1989, pp. 81-102.
6. Sridhar, B.; and Phatak, A. V.: Simulation and Analysis of Image-Based Navigation System for Rotorcraft Low Altitude Flight. Proceedings of AHS National Specialists' Meeting Automation Applications of Rotorcraft, Atlanta, GA, April 4-6, 1988. To be published in IEEE Transactions on Systems, Man, and Cybernetics.
7. Sridhar, B.; Cheng, V. H. L.; and Phatak, A. V.: Kalman Filter Based Range Estimation for Autonomous Navigation using Imaging Sensors. Proceedings of the 11th IFAC Symposium on Automatic Control in Aerospace, Tsukuba, Japan, July 89.

8. Sridhar, B.; Suorsa, R.; and Smith, P.: Vision Based Techniques for Rotorcraft Low Altitude Flight, Intelligent Robotics: Proceedings of the International Symposium on Intelligent Robotics, Bangalore, India, Jan. 91. To appear in the J. Robotics, May 92.
9. Hussien, B.; and Soursa, R.: Clustering Methods for Removing Outliers from Vision-Based Range Estimates. Proceedings of the SPIE, Intelligent Robotics, x. Boston, Mass., Nov. 11-17, 1991.
10. Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; and Teller, E.: Equation of State Calculations by Fast Computing Machines. J. Chemical Physics, vol. 21, no. 6, June 1953, pp. 1087-1092.
11. Kirkpatrick, S.; Gelatt, C. D., Jr.; and Vecchi, M. P.: Optimization by Simulated Annealing. Science, vol. 220, no. 4598, 1983, pp. 671-680.
12. Hecht-Nielsen, R.: Neurocomputing. Addison-Wesley Pub. Co., Reading, Massachusetts, 1988, pp. 192-195.
13. Krozel, J. A.: Search Problems in Mission Planning and Navigation of Autonomous Aircraft. Masters Thesis, Purdue University, Purdue, Indiana, 1988, pp. 128-136.
14. Jeffrey, W.; and Rosner, R.: Optimization Algorithms: Simulated Annealing and Neural Network Processing. The Astrophysical J., vol. 310, Nov. 1, 1986, pp. 473-481.
15. Suorsa, R.; and Sridhar, B.: Validation of Vision-Based Obstacle Detection Algorithms for Low Altitude Flight. SPIE International Symposium on Advances in Intelligent Systems, Boston, MA, Nov. 1990.
16. Smith, P. N.: Flight Data Acquisition Methodology for Validation of Passive Ranging Algorithms for Obstacle Avoidance. NASA TM-102809, Ames Research Center, Moffett Field, CA 94035-1000, Oct. 1990.
17. Garbow, B. S.; Hillstom, K. E.; and More, J. J.: Documentation for MINPACK subroutine LMDIF1. Argonne National Lab., Mar. 1980.
18. Duda, R. O.; and Hart, P. E.: Pattern Classification and Scene Analysis. John Wiley, New York, 1973.
19. Bohachevsky, I. O.; Johnson, M. E.; and Stein, Myron L.: Generalized Simulated Annealing for Function Optimization. Technometrics, vol. 28, no. 3, Aug. 1986, pp. 209-217.
20. Zahn, C. T.: Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. IEEE Trans. on Computers, vol. C-20, no. 1, Jan. 1971, pp. 68-86.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 1993	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Discrete Range Clustering Using Monte Carlo Methods			5. FUNDING NUMBERS 505-64-36	
6. AUTHOR(S) G. B. Chatterji and B. Sridhar				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Ames Research Center Moffett Field, CA 94035-1000			8. PERFORMING ORGANIZATION REPORT NUMBER A-93044	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-104004	
11. SUPPLEMENTARY NOTES Point of Contact: G. B. Chatterji, Ames Research Center, MS 210-9, Moffett Field, CA 94035-1000 (415) 604-5983				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category - 04			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) For automatic obstacle avoidance guidance during rotorcraft low altitude flight a reliable model of the nearby environment is needed. Such a model may be constructed by applying surface fitting techniques to the dense range map obtained by active sensing using radars. However, for covertness, passive sensing techniques using electro-optic sensors are desirable. As opposed to the dense range map obtained via active sensing, passive sensing algorithms produce reliable range at sparse locations and, therefore, surface fitting techniques to fill the gaps in the range measurement are not directly applicable. Both for automatic guidance and as a display for aiding the pilot, these discrete ranges need to be grouped into sets which correspond to objects in the nearby environment. The focus of this paper is on using Monte Carlo methods for clustering range points into meaningful groups. One of the aims of the paper is to explore whether Simulated Annealing methods offer significant advantage over the basic Monte Carlo method for this class of problems. We compare three different approaches and present results of application of these algorithms to a laboratory image sequence and a helicopter flight sequence.				
14. SUBJECT TERMS Clustering, Simulated annealing, Helicopter guidance			15. NUMBER OF PAGES 13	
			16. PRICE CODE A02	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	